Bayesian Class-Matched Multinet Classifier

Yaniv Gurwicz and Boaz Lerner

Pattern Analysis and Machine Learning Lab
Department of Electrical & Computer Engineering
Ben-Gurion University, Beer-Sheva 84105, Israel
{yanivg, boaz}@ee.bgu.ac.il

Abstract. A Bayesian multinet classifier allows a different set of independence assertions among variables in each of a set of local Bayesian networks composing the multinet. The structure of the local network is usually learned using a joint-probability-based score that is less specific to classification, i.e., classifiers based on structures providing high scores are not necessarily accurate. Moreover, this score is less discriminative for learning multinet classifiers because generally it is computed using only the class patterns and avoiding patterns of the other classes. We propose the Bayesian class-matched multinet (BCM^2) classifier to tackle both issues. The BCM^2 learns each local network using a detection-rejection measure, i.e., the accuracy in simultaneously detecting class patterns while rejecting patterns of the other classes. This classifier demonstrates superior accuracy to other state-of-the-art Bayesian network and multinet classifiers on 32 real-world databases.

1 Introduction

Bayesian networks (BNs) excel in knowledge representation and reasoning under uncertainty [1]. Classification using a BN is accomplished by computing the posterior probability of the class variable conditioned on the non-class variables. One approach is using Bayesian multinets. Representation by a multinet explicitly encodes asymmetric independence assertions that cannot be represented in the topology of a single BN using a several local networks that each represents a set of assertions for a different state of the class variable [2]. Utilizing these different independence assertions, the multinet simplifies graphical representation and alleviates probabilistic inference in comparison to the BN [2]-[4]. However, although found accurate at least as other BNs [3], [4], the Bayesian multinet has two flaws when applied to classification. The first flaw is the usual construction of a local network using a joint-probability-based score [4], [5] which is less specific to classification, i.e., classifiers based on structures providing high scores are not necessarily accurate in classification [4], [6]. The second flaw is that learning a local network is based on patterns of only the corresponding class. Although this may approximate the class data well, information discriminating between the class and other classes may be discarded, thus undermining the selection of the structure that is most appropriate for classification.

We propose the Bayesian class-matched multinet (BCM^2) classifier that tackles both flaws of the Bayesian multinet classifier (BMC) by learning each local network

D.-Y. Yeung et al. (Eds.): SSPR&SPR 2006, LNCS 4109, pp. 145 – 153, 2006.

using a detection-rejection score, which is the accuracy in simultaneously detecting and rejecting patterns of the corresponding class and other classes, respectively. We also introduce the *t*BCM^2 which learns a structure based on a tree-augmented naïve Bayes (TAN) [4] using the SuperParent algorithm [7]. The contribution of the paper is three fold. First is the suggested discrimination-driven score for learning BMC local networks. Second is the use of the entire data, rather than only the class patterns for training each of the local networks. Third is the incorporation of these two notions into an efficient and accurate BMC (i.e., the *t*BCM^2) that is found superior to other state-of-the-art Bayesian network classifiers (BNCs) and BMCs on 32 real-world databases.

Section 2 of the paper describes BNs and BMCs. Section 3 presents the detection-rejection score and BCM^2 classifier, while Section 4 details experiments to compare the BCM^2 to other BNCs and BMCs and their results. Section 5 concludes the work.

2 Bayesian Networks and Multinet Classifiers

A BN model B for a set of n variables $\boldsymbol{X}=\{X_1,\ldots,X_n\}$, having each a finite set of mutually exclusive states, consists of two main components, $B=(G,\boldsymbol{\Theta})$. The first component G is the model structure that is a directed acyclic graph (DAG) since it contains no directed cycles. The second component is a set of parameters $\boldsymbol{\Theta}$ that specify all of the conditional probability distributions (or densities) that quantify graph edges. The probability distribution of each $X_i \in \boldsymbol{X}$ conditioned on its parents in the graph $\boldsymbol{Pa}_i \subseteq \boldsymbol{X}$ is $P(X_i=x_i | \boldsymbol{Pa}_i) \in \boldsymbol{\Theta}$ when we use X_i and $\boldsymbol{Pa}_i$ to denote the ith variable and its parents, respectively, as well as the corresponding nodes.

The joint probability distribution over $\boldsymbol{X}$ given a structure G that is assumed to encode this probability distribution is given by [1]

$$P(\boldsymbol{X}=\boldsymbol{x} \mid G)=\prod_{i=1}^{n} P(X_i=x_i \mid \boldsymbol{Pa}_i, G) \tag{1}$$

where $\boldsymbol{x}$ is the assignment of states (values) to the variables in $\boldsymbol{X}$, x_i is the value taken by X_i, and the terms in the product compose the required set of local conditional probability distributions $\boldsymbol{\Theta}$ quantifying the dependence relations. The computation of the joint probability distribution (as well as related probabilities such as the posterior) is conditioned on the graph. A common approach is to learn a structure from the data and then estimate its parameters based on the data frequency count. In this study, we are interested in structure learning for the local networks of a BMC.

A BN entails that the relations among the domain variables be the same for all values of the class variable. In contrast, a Bayesian multinet allows different relations, i.e., (in)dependences for one value of the class variable are not necessarily those for other values. A BMC [2]-[5], [8], [9] is composed of a set of local BNs, $\{B_1,\ldots,B_{|C|}\}$, each corresponds to a value of the $|C|$ values of the class node C. The BMC can be viewed as generalization of any type of BNC when all local networks of the BMC have the same structure of the BNC [4]. Although a local network must be searched for each class, the BMC is generally less complex and more accurate than a BNC. This is because usually each local network has a lower number of nodes than the

BNC, as it is required to model a simpler problem. The computational complexity of the BMC is usually smaller and its accuracy higher than those of the BNC since both the complexity of structure learning and number of probabilities to estimate increase exponentially with the number of nodes in the structure [2].

A BMC is learned by partitioning the training set into sub-sets according to the values of the class variable and constructing a local network B_k for $\boldsymbol{X}$ for each class value $C=C_k$ using the kth sub-set. This network models the kth local joint probability distribution $P_{B_k}(\boldsymbol{X})$. A multinet is the set of local BNs $\{B_1,\ldots,B_{|C|}\}$ that together with the prior $P(C)$ on C classify a pattern $\boldsymbol{x}=\{x_1,\ldots,x_n\}$ by choosing the class $C_K \ \forall K \in [1,|C|]$ maximizing the posterior probability

$$C_K = \arg\max_{k\in[1,|C|]} \left\{ P(C=C_k \mid \boldsymbol{X}=\boldsymbol{x}) \right\}, \tag{2}$$

where

$$P(C=C_k \mid \boldsymbol{X}=\boldsymbol{x}) = \frac{P(C=C_k, \boldsymbol{X}=\boldsymbol{x})}{P(\boldsymbol{X}=\boldsymbol{x})} = \frac{P(C=C_k)P_{B_k}(\boldsymbol{X}=\boldsymbol{x})}{\sum_{i=1}^{|C|} P(C=C_i)P_{B_i}(\boldsymbol{X}=\boldsymbol{x})}\,. \tag{3}$$

In the Chow-Liu multinet (CL multinet) [4], the local network B_k is learned using the kth sub-set and based on the Chow-Liu (CL) tree [10]. This maximizes the log-likelihood [4], which is identical to minimizing the KL divergence between the estimated joint probability distribution based on the network P_{B_k} and the empirical probability distribution for the sub-set $\hat{P}_k$ [5],

$$\mathrm{KL}(\hat{P}_k, P_{B_k}) = \sum_{\boldsymbol{x}} \hat{P}_k(\boldsymbol{X}=\boldsymbol{x}) \cdot \log\left[\frac{\hat{P}_k(\boldsymbol{X}=\boldsymbol{x})}{P_{B_k}(\boldsymbol{X}=\boldsymbol{x})}\right]. \tag{4}$$

Thus, the CL multinet induces a CL tree to model each local joint probability distribution and employs (2) to perform classification. Further elaborations to the construction of the CL tree may be found in [3]. Also we note that the CL multinet was found superior in accuracy to the naïve Bayes classifier (NBC) and comparable to the TAN [4]. Other common BMCs are the mixture of trees model [9], the recursive Bayesian multinet (RBMN) [8] and the discriminative CL tree (DCLT) BMC [5].

3 The Bayesian Class-Matched Multinet Classifier

We suggest the Bayesian class-matched multinet (BCM^2) that learns each local network using the search-and-score approach. The method searches for the structure maximizing a discrimination-driven score that is computed using training patterns of all classes. Learning a local network in a turn rather than both networks simultaneously has computational benefit regarding the number of structures that need to be considered. First we present the discrimination-driven score and then the tBCM^2 that is a classifier based on the TAN [4] and searched using the SuperParent algorithm [7].

The BCM^2 Score. We first make two definitions: (a) a pattern $\boldsymbol{x}$ is *native* to class C_k if $\boldsymbol{x} \in C_k$ and (b) a pattern $\boldsymbol{x}$ is *foreign* to class C_k if $\boldsymbol{x} \in C_j$ where $j \in [1,|C|]$ and $j \neq k$. We partition the dataset D into test (D_{ts}) and training (D_{tr}) sets, the latter is further divided into internal training set T used to learn candidate structures and a validation set V used to evaluate these structures. Each training pattern in D_{tr} is labeled for each local network B_k as either native or foreign to class C_k depending on whether it belongs to C_k or not, respectively. In each iteration of the search for the most accurate structure, the parameters of each candidate structure are learned using T in order to construct a classifier that can be evaluated using a discrimination-driven score on the validation set. After selecting a structure, we update its parameters using the entire training set (D_{tr}) and repeat the procedure for all other local networks. The derived BCM^2 can be then tested using (2).

The suggested score evaluates a structure using the ability of a classifier based on this structure in detecting native patterns and rejecting foreign patterns. The score S_x for a pattern $\boldsymbol{x}$ is determined based on the maximum a posteriori probability, i.e.,

$$S_{\boldsymbol{x}} = \begin{cases} 1, \mathit{if}\ \{P(C=C_k \mid \boldsymbol{X}=\boldsymbol{x}_n^k) \geq P(C \neq C_k \mid \boldsymbol{X}=\boldsymbol{x}_n^k)\}\ \mathit{or}\ \{P(C \neq C_k \mid \boldsymbol{X}=\boldsymbol{x}_f^k) > P(C=C_k \mid \boldsymbol{X}=\boldsymbol{x}_f^k)\} \\ 0, \mathit{if}\ \{P(C=C_k \mid \boldsymbol{X}=\boldsymbol{x}_n^k) < P(C \neq C_k \mid \boldsymbol{X}=\boldsymbol{x}_n^k)\}\ \mathit{or}\ \{P(C \neq C_k \mid \boldsymbol{X}=\boldsymbol{x}_f^k) \leq P(C=C_k \mid \boldsymbol{X}=\boldsymbol{x}_f^k)\} \end{cases}, \quad (5)$$

where $\boldsymbol{x}_n^k$ and $\boldsymbol{x}_f^k$ are native and foreign patterns to C_k, respectively. The first line in (5) represents correct detection (classification of a native pattern to C_k) or correct rejection (classification of a foreign pattern to a class other than C_k), whereas the second line represents incorrect detection of a native pattern or incorrect rejection of a foreign pattern. By identifying *TP* (true positive) as the number of correct detections and *TN* (true negative) as the number of correct rejections made by a classifier on all the $|V|$ validation patterns in V, we define the detection-rejection measure (*DRM*)

$$DRM = \frac{\sum_{x \in V} S_x}{|V|} = \frac{(TP + TN)}{|V|}, \qquad DRM \in [0,1]. \quad (6)$$

That is, for each local network and each search iteration, we select the structure that the trained classifier based on this structure simultaneously detects native patterns and rejects foreign patterns most accurately. Both correct detection and correct rejection contribute equally to the score although any other alternative is possible.

TAN-Based BCM^2. We propose a TAN-based BCM^2 (tBCM^2) that utilizes the *DRM* and SuperParent algorithm searching the TAN space. The SuperParent (SP) algorithm has reduced computational cost compared to hill-climbing search (HCS) and it expedites the search [7]. In each iteration, we determine the best edge to add to a structure by finding a good parent and then the best child for this parent.

Following [7] we define: (a) an *Orphan* is a node without a parent other than the class node, (b) a *SuperParent* (SP) is a node extending edges to all orphans simultaneously (as long as no cycles are formed) and (c) a *FavoriteChild* (FC) of an SP is the orphan amongst all orphans that when connected to the SP provides a structure having the highest value of the *DRM*. We initialize the search for each local network using the NBC structure and employ the value of *DRM* it provides as the current *DRM* value. Each iteration of the search comprises of two parts. First, we make each node an SP in turn and choose the SP that if added to the structure would provide the highest value of the *DRM*. Second, we find the FC for this SP and add the edge between them to the structure if this edge increases the current value of the *DRM*. We update the current value of the *DRM* and continue the search as long as the *DRM* value increases and more than one orphan remains unconnected to an SP. Since in each iteration we connect one variable at the most, the maximum number of iterations and edges that can be added to the initial structure is n-1 (yielding the TAN structure). We repeat this procedure for all $|C|$ local networks terminating with the tBCM^2, as is exemplified in the following pseudo code:

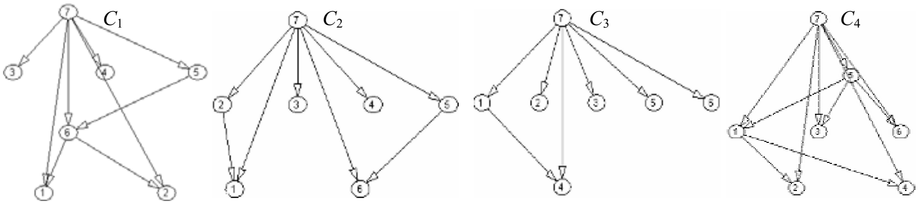
```
1. For $k=1:|C|$     // index of the local network $B_k$
1.1 Start with the NBC structure as the current structure of the kth local network. In
      all stages, use $T$ to learn the structure and $V$ to calculate the structure DRM.
1.2 For $g=1:n-1$      // index of iteration
1.2.1 Find the SP yielding the structure having the highest DRM.
1.2.2 Find the FC for this SP.
1.2.3 If the edge $\text{SP} \rightarrow \text{FC}$ improves the DRM value of the current structure, update
          the structure with this edge and employ the structure as the current structure.
        Else: Return the current structure as the kth local network and go to 1.
1.3 Return the current structure as the kth local network and go to 1.
2. Calculate the parameters of each local network using $D_{tr}$ and return the $t$BCM$^2$.
```

Although both the CL multinet and tBCM^2 learn a multinet based on the TAN, the two algorithms differ in a several main issues. First, the CL multinet is learned using a constraint-based approach [11] based on the CL tree algorithm [10] or an extended version of this algorithm [3], while the tBCM^2 is learned by employing the search-and-score approach [11]. Second, the former algorithm establishes for each class a CL tree that maximizes a joint-probability-based measure, whereas the latter algorithm employs a discrimination-driven score for structure learning. Third, the CL multinet utilizes only the class patterns for learning each local network, whereas the tBCM^2 utilizes all patterns. Fourth, the CL multinet always adds n-1 edges even when some variables are completely independent, while the tBCM^2 stops adding edges when there is no improvement in the score of a local network.

Finally we note that the worst case computational complexity of the tBCM^2 (excluding the cost of parameter learning) is $O(3 \cdot |C| \cdot |V| \cdot n^3/2)$, which incurs if the algorithm does not end before finding the maximum possible number of SPs [12]. As an example, Figure 1 demonstrates the four local networks learned by the tBCM^2 for the UCI repository Car database [13] along with the corresponding *DRM* values.

4 Experimental Results

Between the *DRM* and Classification Accuracy. Since the *DRM* is measured for each local network separately and using the validation set and the classification accuracy is measured for the *t*BCM^2 and the test set, we studied the relation between the two scores. We started the search for each local network with the NBC structure and identified an iteration by the addition of an edge between an SP and its FC. Whenever all the local networks had completed an iteration, we computed the values of *DRM* they achieve, the average *DRM* value and the test accuracy of the *t*BCM^2 that used these networks. We repeated this procedure until all local networks completed learning (i.e., all final structures were found). Networks that completed learning before their counterpart networks, contributed their final *DRM* values to the calculation of the average *DRM* in each following iteration. Figure 2a presents the relation between the average *DRM* value of the local networks and the classification accuracy of the *t*BCM^2 for increasing numbers of iterations of the SP algorithm and the UCI repository Nursery database [13]. This database is large (i.e., providing reliable results) and has relatively many variables that introduce numerous possible edge additions in each search iteration, thereby the database enables testing structure



DRM=0.9297 *DRM*=0.9271 *DRM* =0.9531 *DRM* =0.9844

Fig. 1. The four local networks and associated *DRM* values of the *t*BCM^2 for the Car database

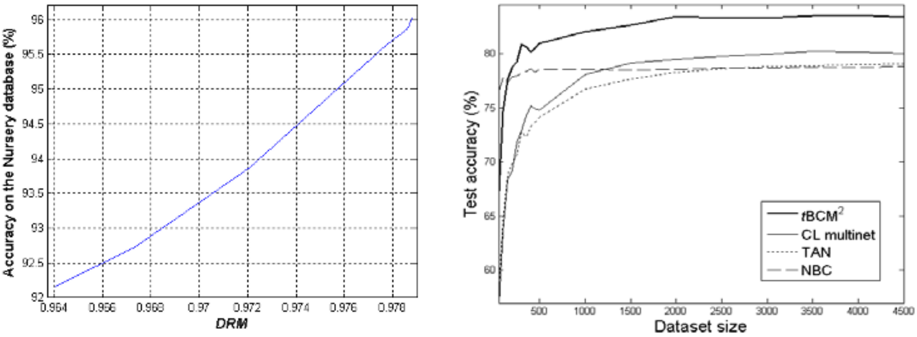


Fig. 2. (a) The relation between the average *DRM* and the *t*BCM^2 classification accuracy for increasing numbers of iterations of the SP algorithm and the UCI Nursery database. (b) Learning curves for the *t*BCM^2, CL multinet, TAN and NBC for the Waveform-21 database.

learning extensively. The figure shows that the classification accuracy increases monotonically with the average *DRM* value.

Learning Curves. Figure 2b presents learning curves for the *t*BCM^2, CL multinet, TAN and NBC for the large UCI repository Waveform-21 database [13]. Each of ten random replications of the database was partitioned into ten sets. One set was reserved for the test, and the other nine sets were added incrementally to the training set. Each classifier was trained using the increased-size training set and tested on the same test set following each increase. The accuracy was repeatedly measured for all data replications and averaged. Figure 2b demonstrates that the NBC and CL multinet have, respectively, the smallest and largest sensitivity to the sample size. The former classifier has lesser sensitivity since it needs to estimate only few parameters so even a small sample size provides the classifier its asymptotic accuracy. The *t*BCM^2 is less sensitive than the CL multinet for two reasons. First, the *t*BCM^2 may have fewer edges for each of its local networks than the CL multinet (Section 3) and therefore it needs to estimate less parameters. Second, the *t*BCM^2 utilizes all the data whereas the CL multinet employs only the class data. In addition we note that except for a very small sample size, the *t*BCM^2 is superior to all other classifiers for this database. Similar conclusions are drawn for most of the other databases.

Classification Accuracy. Table 1 demonstrates the superior classification accuracy of the *t*BCM^2 in comparison to the NBC, TAN, CL multinet and RBMN for 32 databases of the UCI repository. Out of the databases, the *t*BCM^2 accomplishes higher accuracy than the CL multinet on 24 databases, identical accuracy on 3 databases and inferior accuracy on 5 databases. It achieves higher accuracy than the TAN on 28 databases and inferior accuracy on 4 databases. The *t*BCM^2 also outperforms the NBC on 90% of the databases. Twenty-two databases are tested using CV10 and the remaining (large) databases using holdout. On the former databases, the *t*BCM^2 reaches higher accuracy than the CL multinet classifier on 16 of the databases with statistical significance of 95% (t-test with α=0.05) on 12 of the databases and the CL multinet classifier achieves higher accuracy than the *t*BCM^2 on 4 of the databases without statistical significance for none of them. Also for these 22 databases, the *t*BCM^2 accomplishes higher accuracy than the TAN on 18 of the databases with statistical significance of 95% (α=0.05) for 13 of them and the TAN achieves higher accuracy than the *t*BCM^2 on 4 of the databases with statistical significance of 95% (α=0.05) for 1 of the databases.

In addition, Table 1 exemplifies the *t*BCM^2 superiority to the RBMN [8] for those databases for which results are provided. Also, we compare the *t*BCM^2 to the DCLT algorithm [5] for the only two databases for which results are given in [5]. We find for the Hepatitis database accuracies of 89.25% and 90.4% and for the Voting database accuracies of 92.18% and 93.97% for the DCLT and *t*BCM^2 classifiers, respectively. Finally, Table 1 presents also the average classification accuracies of the inspected methods over all 32 databases. The table shows that the *t*BCM^2 (89.64%) is superior on average to the NBC (85.74%), TAN (87.41%) and CL multinet (87.45%).

Table 1. Classification accuracies of the *t*BCM^2 and other classifiers on 32 databases from [13]. Bold font represents the highest accuracy for a database.

Database	NBC	TAN	CL multinet	RBMN	*t*BCM^2
Adult	83.61	85.83	85.11	NA	**87.33**
Australian	85.36 (±2.14)	84.15 (±2.17)	85.22 (±2.09)	85.21	**88.38 (±2.32)**
Balance	**91.85 (±2.54)**	85.44 (±2.07)	84.63 (±1.81)	NA	90.88 (±1.03)
Breast	97.51 (±0.94)	96.12 (±1.99)	96.34 (±1.00)	95.75	**98.24 (±1.14)**
Car	85.71 (±2.33)	89.81 (±1.89)	**94.10 (±0.98)**	93.06	93.92 (±0.81)
Cmc	51.66 (±2.97)	52.00 (±1.10)	50.85 (±1.82)	NA	**52.85 (±1.65)**
Corral	85.06 (±4.59)	96.06 (±2.44)	99.23 (±2.93)	NA	**100.0 (±0.00)**
Crx	85.98 (±1.85)	85.67 (±2.72)	86.14 (±2.79)	**90.05**	88.89 (±2.59)
Cytogenetic	77.94	81.14	80.30	NA	**82.87**
Flare	79.82 (±1.66)	82.54 (±1.17)	82.55 (±0.94)	**86.87**	83.35 (±1.21)
Hayes	**81.88 (±4.25)**	75.00 (±3.27)	63.13 (±4.86)	NA	80.63 (±3.13)
Hepatitis	85.23 (±1.27)	86.01 (±1.78)	86.54 (±2.00)	NA	**90.40 (±1.52)**
Ionosphere	91.16 (±2.34)	91.44 (±2.57)	**93.92 (±2.01)**	NA	93.03 (±2.69)
Iris	93.67 (±2.99)	93.33 (±2.16)	93.33 (±2.16)	NA	**95.83 (±2.21)**
Krkp (Chess)	87.32	92.31	93.02	94.18	**95.03**
Led-7	74.41	73.76	73.10	NA	**75.89**
Lymphography	83.19 (±3.93)	84.57 (±5.47)	79.81 (±5.05)	NA	**85.57 (±5.16)**
Mofn-3-7-10	85.05 (±1.80)	91.06 (±2.01)	90.63 (±2.46)	90.53	**94.43 (±2.30)**
Monks	96.39 (±1.68)	98.73 (±1.41)	**98.92 (±1.09)**	NA	**98.92 (±1.09)**
Mushroom	97.40	99.47	99.47	NA	**100**
Nursery	89.17	91.09	93.89	91.06	**96.03**
Pendigit	85.72	94.32	**96.62**	NA	96.04
Segment	91.34 (±0.83)	94.09 (±1.04)	94.42 (±1.23)	89.35	**96.13 (±1.23)**
Shuttle	98.45	99.61	**99.92**	97.21	**99.92**
Splice (DNA)	96.33	89.65	96.74	87.52	**97.98**
Tic Tac Toe	69.62 (±1.96)	**75.07 (±2.64)**	73.07 (±2.41)	NA	72.65 (±1.45)
Tokyo	91.45 (±1.82)	92.01 (±2.19)	92.39 (±1.55)	NA	**93.94 (±1.98)**
Vehicle	62.42 (±2.67)	70.82 (±2.51)	69.93 (±2.91)	**73.64**	68.54 (±2.65)
Voting	90.96 (±2.62)	93.99 (±2.16)	93.97 (±2.46)	**96.55**	93.97 (±2.46)
Waveform-21	78.60	78.94	79.69	77.79	**83.82**
Wine	98.27 (±1.65)	98.03 (±1.55)	98.27 (±1.65)	NA	**98.98 (±1.21)**
Zoo	92.00 (±4.66)	**95.08 (±4.25)**	93.09 (±5.03)	NA	94.08 (±4.17)
Average	85.74	87.41	87.45	**---**	**89.64**

5 Summary and Concluding Remarks

We propose the *t*BCM^2 which is a multinet classifier that learns each local network based on a detection-rejection measure, i.e., the accuracy in simultaneously detecting and rejecting, respectively, the corresponding class and other class patterns. The *t*BCM^2 uses the SuperParent algorithm to learn for each local network a TAN having only augmented edges that increase the classifier accuracy. Evaluated on 32 real-world databases, the *t*BCM^2 demonstrates on average superiority to the NBC, TAN, CL multinet and RBMN classifiers. The advantage of the *t*BCM^2 to the TAN is related to the facts that the former classifier is a multinet that is learned using a discrimination-driven score, and the advantage of the *t*BCM^2 to the CL multinet is attributed to the score of the former and the facts that it usually learns a smaller number of parameters and use the whole data for training.

In further work, we will make parameter learning discriminative rather than generative and apply the BCM^2 to less restricted structure spaces, such as augmented naïve and general Bayesian networks.

Acknowledgment. This works was supported in part by the Paul Ivanier Center for Robotics and Production Management, Ben-Gurion University, Beer-Sheva, Israel.

References

1. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan-Kaufmann, San-Francisco (1988)
2. Geiger, D., Heckerman, D.: Knowledge representation and inference in similarity networks and Bayesian multinets. Artificial Intelligence 82 (1996) 45-74
3. Cheng, J., Greiner, R.: Learning Bayesian belief network classifiers: Algorithms and system. In Proc. 14th Canadian Conf. on Artificial Intelligence (2001) 141-151
4. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Machine Learning 29 (1997) 131-163
5. Huang, K., King, I., Lyu, M. R.: Discriminative training of Bayesian Chow-Liu multinet classifier. In Proc. Int. Joint Conf. Neural Networks (2003) 484-488
6. Kontkanen, P., Myllymaki, P., Sliander, T., Tirri, H.: On supervised selection of Bayesian networks. In Proc. 15th Conf. on Uncertainty in Artificial Intelligence (1999) 334-342
7. Keogh, E.J., Pazzani, M.J.: Learning the structure of augmented Bayesian classifiers. Int. J. on Artificial Intelligence Tool*s* 11 (2002) 587-601
8. Pena, J.M., Lozano, J.A., Larranaga, P.: Learning recursive Bayesian multinets for data clustering by means of constructive induction. Machine Learning 47 (2002) 63-89
9. Meila, M., Jordan, M.I.: Learning with mixtures of trees. J. of Machine Learning Research 1 (2000) 1-48
10. Chow, C.K., Liu, C.N.: Approximating discrete probability distributions with dependence trees. IEEE Trans. Info. Theory 14 (1968) 462-467
11. Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction and Search. 2nd edn. MIT Press, Cambridge MA (2000)
12. Gurwicz, Y.: Classification using Bayesian multinets. M.Sc. Thesis. Ben-Gurion University, Israel (2004)
13. Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases. http://www.ics.uci.edu/~mlearn/MLRepository.html, 1998